AFRL-RX-WP-TR-2010-4273

SUPPLY CHAIN RISK MODELING AND SIMULATION USING FLOW EQUIVALENT SERVERS

Charlie Stirk

CostVision, Inc.

JULY 2010
Final Report

Approved for public release; distribution unlimited.

See additional restrictions described on inside pages

STINFO COPY

AIR FORCE RESEARCH LABORATORY
MATERIALS AND MANUFACTURING DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7750
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the USAF 88th Air Base Wing (88 ABW) Public Affairs Office (PAO) and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RX-WP-TR-2010-4273 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//Signature//_____	//Signature//_____
BRENCH L. BODEN, Program Manager	SCOTT M. PEARL, Chief
AFRL/RXMT	AFRL/RXMT

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

*Disseminated copies will show "//signature//" stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
July 2010	Final	30 September 2008 – 30 June 2010

4. TITLE AND SUBTITLE
SUPPLY CHAIN RISK MODELING AND SIMULATION USING FLOW EQUIVALENT SERVERS

5a. CONTRACT NUMBER
FA8650-08-C-5709

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
63680D

6. AUTHOR(S)
Charlie Stirk

5d. PROJECT NUMBER
6036

5e. TASK NUMBER
01

5f. WORK UNIT NUMBER
M0112200

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

CostVision, Inc.
1472 North Street
Boulder, CO 80304-3512

Georgia Tech University
Rockwell Collins

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Materials and Manufacturing Directorate
Wright-Patterson Air Force Base, OH 45433-7750
Air Force Materiel Command
United States Air Force

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RXMT

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RX-WP-TR-2010-4273

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
Report contains color. PAO Case Number: 88ABW-2011-2144; Clearance Date: 12 Apr 2011.

14. ABSTRACT ***(Maximum 200 words)***

The purpose of the Flow Equivalent Servers (FES) project is to develop a reliable but simple model to replace detailed simulation of individual suppliers in supply chains. The end goal of these new capabilities is to provide robust analysis for DoD acquisition supply networks with a reduction in analysis cost and time.

15. SUBJECT TERMS
supply chain, risk model, flow equivalent server

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			Brench L. Boden
Unclassified	Unclassified	Unclassified	SAR	46	19b. TELEPHONE NUMBER *(Include Area Code)* (937) 904-4360

Table of Contents

Section Page

Table of Figures

Table of Tables

Executive Summary

The purpose of the Flow Equivalent Servers (FES) project is to develop a reliable but simple model to replace detailed simulation of individual suppliers in supply chains. The end goal of these new capabilities is to provide robust analyses for DoD acquisition supply networks with a reduction in analysis cost and time.

Due to the current limits of production theory, we have to develop a novel approach to achieve this goal. A new model is derived for approximating the cycle time and WIP behavior of a factory simulation, based on newly observed properties in general queuing networks. Using these properties, the newly developed models outperform existing approaches and give very small approximation errors.

In consideration of the data availability and confidentiality issues likely to arise in supply network analysis, two approaches are examined, referred to as black box and white box. The black box approach assumes we do not know the details of the simulation models but only the input and output data. The white box approach assumes we can examine the details of the simulation models, therefore, more accurate approximate results can be achieved.

In Phase I, we only focus on the single product scenario. Therefore, if there are multiple products, we only calculate the total cycle time instead of the cycle time for each product. When there are multiple products, the approximate model for the cycle time of each product is not available.

1) Introduction

The purpose of the Flow Equivalent Servers (FES) project is to develop a reliable but simple model to replace detailed simulation of individual suppliers in supply chains. The end goal of these new capabilities is to provide robust analysis for DoD acquisition supply networks with a reduction in analysis cost and time.

For DoD acquisition supply networks that have difficulty predicting and analyzing risk in material flows and schedule integration, this project develops the basic methodology of FES, a statistically-based model that can replace a single facility simulation in a hierarchical simulation of a supply chain. Unlike previous approaches that are either static or require manual and special-purpose model integration, the FES approach is dynamic, flexible, manageable, supports different tools, and could be automated.

Initial results show FES can be a very accurate tool. In this Phase I effort, the FES team has shown the technical feasibility of the approach through the following tasks.

- Developed and tested FES models for suppliers that accurately reproduce their simulated or observed schedule and capacity performance.

- Demonstrated feasibility to extract FES from representative manufacturing simulation models.

- Defined a neutral data format for the exchange of FES models between discrete event simulation software packages.

- Defined the requirements for an FES model configuration management and linking system based on the requirements of multiple DoD acquisition programs.

This Phase I final report describes the accomplishments made. Specifically, this paper will describe the approaches investigated, the technical hurdles encountered, and recommendations by the team for future efforts.

2) Problem Statement

Planning and managing major defense acquisition programs (DAPs) requires balancing and synchronizing design and production across a network of distributed activities performed by independent entities. The complexity of these networks makes the prediction of time, cost, and risk very difficult. The best known prediction method today is to create discrete event simulation models of the manufacturing processes and the supply chain.

However, these models at the manufacturing process and supply chain levels are rarely integrated for several reasons. Developing integrated simulation models is expensive and time consuming; several efforts that we have evaluated took several expert developers at least six months to build. In addition, integrated simulation modeling requires the participating companies to divulge key capability and capacity information they may consider proprietary. If the models are not designed to be integrated from the outset, it is often impossible because their simulation conventions may be incompatible. For instance, a manufacturing simulation may model individual products, and a supply chain simulation may model shipments of many products in a collection.

The most common approach is to value stream map and then simulate a manufacturing process. Then the manufacturing processes are stress tested against several static scenarios of the Integrated Master Schedule for the program with different supply chain configurations and production volume levels. However, this method does not explore all the risks because it does not reveal dynamic affects in the supply chain such as how much safety stock is needed to buffer against missed deliveries or how surges can lead to increased cycle times due to capacity constraints. Thus, there is a need for a new approach to supply chain analysis that allows supply chain partners to quickly and reliably reach a program plan that optimizes the complex trade-offs between costs, time, and risks.

The participants in DAP supply chains often create high fidelity simulation models of their own facilities and operations. This is becoming more common as simulation skills and the benefits of simulations become more widely known. On some programs, manufacturing process is a requirement in the contract. It is also a best practice that is recognized in the Manufacturing Readiness Levels (MRLs). For instance, the MRL Matrix contains a sub-thread on Modeling & Simulation (Product & Process) that contains the following criteria in Table 1.

MRL4	MRL 5	MRL 6
Initial simulation models (product or process) developed at the component level.	Initial simulation models developed at the sub-system or system level.	Simulation models used to determine system constraints and identify improvement opportunities.

Table 1. Manufacturing Readiness Level Matrix criteria for the sub-thread on Modeling & Simulation (Product & Process). [1]

[1] MRL Matrix V10 4, spreadsheet available from DAU MRL web site, 2010.

Component level simulations can either be continuous physics-based simulations or higher level discrete event simulations. These simulation models are often developed separately for each component and at the sub-system or system level. The challenge is how to integrate these simulations at different levels, which is often across the supply chain.

Two observations from the National Academy report, "Modeling and Simulation in Manufacturing and Defense Acquisition,"[2] summarize the difficulty in using these simulation models in an integrated way:

- "Currently, a state of the art, standardized external model representation is lacking", and
- "Modeling languages do not adequately support the structuring of large, complex models and the process of model evolution."

In other words, even if supply chain partners have simulation models of their own manufacturing and logistics operations, these models are not likely to incorporate external representations necessary for transparent interoperability. Moreover, the simulation technologies currently available do not adequately support their integration or the continued evolution of an integrated model. Both manufacturing and supply chain modeling and simulation need a new modeling technology and a compatible and neutral format for interoperability and for managing complex federated models.

Nevertheless, these manufacturing process simulation models provide a basis for analyzing a proposed or operational supply chain. The challenge is to exploit these existing simulation models without requiring supply chain partners to expose their proprietary details of capacity, capability, quality, etc. There are two approaches to meeting this challenge: (1) use the simulation models directly, by federating them or (2) use the simulation models indirectly, by creating neutral format approximations that can be integrated without revealing proprietary details. The FES project is focused on the second alternative.

The direct approach to federating simulations, using a technology such as HLA[3] (High Level Architecture), has been used with considerable success to federate battle space simulations. There have been some publications dealing with HLA and manufacturing simulation, although they tend to be either theoretical[4] or provide only small examples. There is very limited information on HLA applied to supply chains, and the typical publication describes a problem with very limited scope[5]. There is little evidence that HLA has been applied successfully to large scale manufacturing or supply chain simulation federates. This is, in large part, because the

[2] Modeling and Simulation in Manufacturing and Defense Acquisition: Pathways to Success, National Research Council, ISBN: 0-309-56614-2 (2002).

[3] U.S. Defense Modeling and Simulation Office (2001). RTI 1.3-Next Generation Programmer's Guide Version 4. U.S. Department of Defense.

[4] Alvarado, J. R., R Velez Osuna, and R. Tuokko, "Distributed simulation in manufacturing using high level architecture," in Micro-Assembly Technologies and Applications, Springer Boston, 2008.

[5] Gan, B. P., L. Liu, S. Jain, S. Turner, W. Cai, and W. Hsu, "Distributed supply chain simulation across enterprise boundaries," Winter Simulation Conference, 2000.

intensity of interactions among the federates and the discrete event nature of interactions make it very difficult to synchronize the federates. Federated manufacturing and supply chain simulations are difficult to develop from existing non-HLA-compliant simulations and extremely slow to compute.

The approach proposed here also is a form of federation, not of high fidelity simulation models, but of reduced dimension approximations of those simulation models. The overall FES project aims to demonstrate a methodology for creating the reduced dimension approximations, creating a repository, based on a standard format, and federating the approximations to create lower fidelity but comprehensive supply chain simulations.

The project team envisions an application scenario in which prospective supply chain partners collaborate in negotiating the design of the proposed DAP supply chain. Each partner, using high fidelity simulation models of their own facilities and operations, develops internal resource and operational plans and assesses their contribution to the delivery time, cost, and risk of the product. For a given scenario, each prospective supply chain partner prepares a relatively low fidelity approximation of their high fidelity simulation model, and the resulting set of low fidelity approximations is federated in order to provide an approximation of the overall performance of the supply chain in terms of delivery time, cost, and risk. The goal is to achieve more robust program acquisition plans with reduced cost and reduced risk.

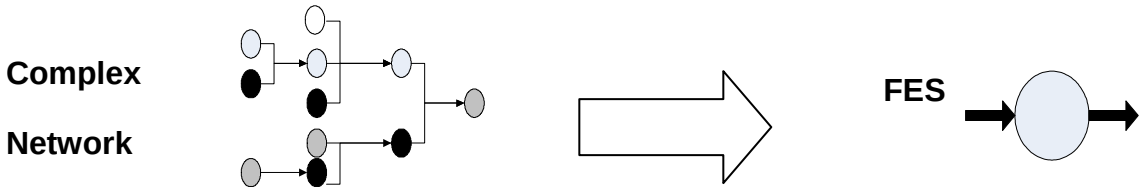


Figure 1. A Flow Equivalent Server is a single server approximation transformed from a complex queuing network.

3) Definition of Terminology

In the field of manufacturing simulation, some terminology is not yet completely standardized. To avoid confusion, this section provides the definitions used in the work reported here.

System bottleneck: the throughput bottleneck which is the workstation with the highest utilization.

2nd system bottleneck: the throughput bottleneck which is the workstation with the 2nd highest utilization.

System utilization: the utilization of the system bottleneck.

Capacity: the maximum throughput rate of a machine.

Service time: the reciprocal of capacity (i.e., if the capacity is 2 jobs/ 480 minute shift, the service time is 240 minutes/job).

Processing time: the time that a job spends on a machine in order to complete its process (may be different from service time due to setups or other interruptions).

Factory Process Time (PTf): cycle time for one product to be produced (zero queuing time).

Variability: variability of a random variable is its variance divided by its mean square.

Residual Standard Error: the quantitative measure for how good the fit is (lower residual is better). It is given by the following formula:

$$Residual\ Standard\ Error = \sqrt{\frac{\sum(observed_i - fitted_i)^2}{n-2}}$$

Maximum Error Percentage: the maximum difference between observed value and fitted value as a percentage of the observed value. It is given by the following formula.

$$Maximum\ Error\ Percentage = Max\left(\frac{|observed_i - fitted_i|}{observed_i} X\ 100\right)$$

Cycle time of a job in an assembly line: since a job in an assembly line is composed of multiple parts, the cycle time of a job is determined by the part with the longest cycle time, where its cycle time is the duration difference between its completion and its start (i.e. released to the production line).

ASIA system: an ASIA system is one in which it is assumed that for the workstations in the system, "all see initial arrivals" (ASIA) directly. For a given tandem queue, an ASIA system results from assuming servers are in parallel rather than in series.

Randomness and synchronization effects: randomness effects and synchronization effects are the two causes of queuing time in practical manufacturing systems, where queuing time is the duration that a customer waits for service.

A randomness effect usually comes from the variation of inter-arrival times and service times. It could also be caused by interruptions. When randomness effects exist, queuing time will grow without limit as the arrival rate approaches the service rate.

When there is no randomness in the system, queuing can be completely avoided if arrival intervals are synchronized with service times. For example, the inter-arrival times are constant 40, and the service times are constant 30.

However, queuing time can still occur when the arrival intervals and service times are not synchronized, even when there is no randomness in the system. This occurs when a manmade control (vs. natural behavior) is exerted on the system to achieve a pre-specified objective. For example, suppose the inter-arrival times are always two constant 20s and then followed by one 80 (e.g., 20, 20, 80, 20, 20 and 80, etc.), and the service times are 30. Although the mean inter-arrival times are 40, there are always queuing times for the last two of every three jobs. This arrival pattern can result from the changeover rules (or dispatching rules, in general) of the upstream machines: it may send downstream machines some jobs continuously and then stop sending for a while. Transfer batches (or parallel process batches) with constant batch size k are another example of this kind of situation, since the inter-arrival times (or service times) can be viewed as k-1 zeros plus a large positive.

Queuing time caused by a synchronization effect is commonly seen in assembly lines. Even if all service times and inter-arrival times are deterministic, a component may still wait for other components in front of an assembly stage. It also can be induced by shift schedules. For example, although all machines work 24 hours a day, some machines need assistance from operators. Operators only work 10 hours a day with one hour break in between (i.e., 5 + 1 + 5). Machines which need operators can keep working until finishing their current jobs even without an operator. But a job has to wait if it arrives at those machines when operators are not available. When there are shift schedules, queuing time can occur even if all service times and arrival intervals are deterministic.

In practice, queuing times are caused by the mix of the randomness and synchronization effects. Usually, queuing time caused by a synchronization effect can be analyzed exactly if there is no randomness in the system. However, the exact analysis becomes difficult if it is a mix of the two. The analysis becomes extremely difficult if it is combined with the non-renewal departure process in general queuing networks.

4) Approaches Investigated

The fundamental challenge in achieving the vision described in the Problem Statement is to develop a suitable method, or collection of methods, for approximating the cost, time, and risk assessments obtained from a high fidelity simulation model. The simulation models of interest all represent manufacturing or logistics processes, thus represent an underlying network of processes and material flows. This network structure is key to the proposed methodology.

There is a large body of research addressing the analysis of processing networks, and a brief summary of the relevant prior research is presented later in Technical Discussion. While this prior research provides valuable insight, it does not directly address the fundamental challenge, because it is primarily focused on exact analysis or approximation of highly abstracted models, under very stringent assumptions. Our goal, however, is to achieve approximation of highly realistic models of actual manufacturing and logistics processes. While it may be possible to exploit the prior research in novel ways, direct application is not an option. A new approach is required.

Our overall approach is based on the following assumptions:

1. Each supply chain partner has or creates a simulation model of the facilities and processes to be assigned to its portion of the proposed DAP supply chain.

2. For each such simulation model, the methods to be developed are applied to create a suitable reduced dimension approximation. There are two possibilities:

   a. The pre-existing simulation is treated as a "black box" so the approximation method has access only to the input parameters and output simulation results; this approach is completely empirical.

   b. The pre-existing simulation is treated as a "white box" in which the internal structure and logic are for the purpose of creating the approximation; this approach combines analytic methods and empirical methods.

3. The resulting approximation is archived in a standard format.

4. Integrated supply chain models are created from these archived individual models.

5. Analysis of time, cost, and risk is performed on the resulting comprehensive reduced dimension model.

6. The methods developed are applicable to any simulation language, but the research and demonstration has used only Arena.

Because the requirement cannot be achieved by the current known approaches, meeting the requirements requires some innovation. The following two sections describe the "black box" approach and the "white box" approach.

4.1) Black Box Approach

It is assumed that the partners in a DAP supply chain each have or can develop simulation models of the resources they commit to the program. The black box approach to developing reduced dimensionality approximations treats each of these simulations as a "black box," i.e., all that can be observed is the input parameters to the simulation and the output statistics from executing the simulation. From these inputs and outputs, we construct a "flow equivalent server" to approximate the high fidelity simulation.

This approach has been developed and tested using simulation models provided by the PDES Model Based Manufacturing group as shown in Section 4.4. The basic process in the black box approach is to exercise the pre-existing simulation model over a range of input rates (equivalently, a range of system utilizations) and then to fit a statistical model to the observed cycle times. The process involves starting with a very low value (to estimate the basic process time) and then iteratively increasing the input rate to the simulation to identify the "saturation" input rate, or maximum system throughput. As the input rate approaches the saturation input rate, the run time required for the simulation model to reach steady state will increase, and managing this iterative process represents an important technical issue to be resolved. The more detailed explanation will be given in Section 5.1.

Essentially, the FES model is a statistical model, and there are several possible modeling strategies. The simplest is to assume the data come from a single station with general arrival time and service time distributions and fit a G/G/m analytic model, e.g., from Factory Physics[6].

$$E(QT_{G/G/m}) \cong \alpha \frac{\rho^{\sqrt{2(m+1)}-1}}{(1-\rho)} \frac{1}{m\mu}.$$

In this case, the number of servers, *m*, and the variability term, (α), rather than being computed from known parameters of the arrival and service time distributions, is treated as a parameter to be estimated by fitting the experimental data. In preliminary research, a model developed by Wu[7] also was used, which gave improved results. Figure 2 illustrates the results from applying the black box approach to one simulation model.

In summary, the black box approach is to select an analytic model, such as G/G/m, and fit this model to the data, where the variability factor is treated as a variable to be determined by the statistical analysis. Likewise, m may be assumed to be 1, or it might also be treated as a variable to be determined by the statistical analysis.

[6] Hopp, W. and M. Spearman (2000). Factory Physics, McGraw Hill.

[7] Wu, K. (2005). "An Examination of Variability and Its Basic Properties for a Factory." IEEE Transactions on Semiconductor Manufacturing, 18(1): 214-221.

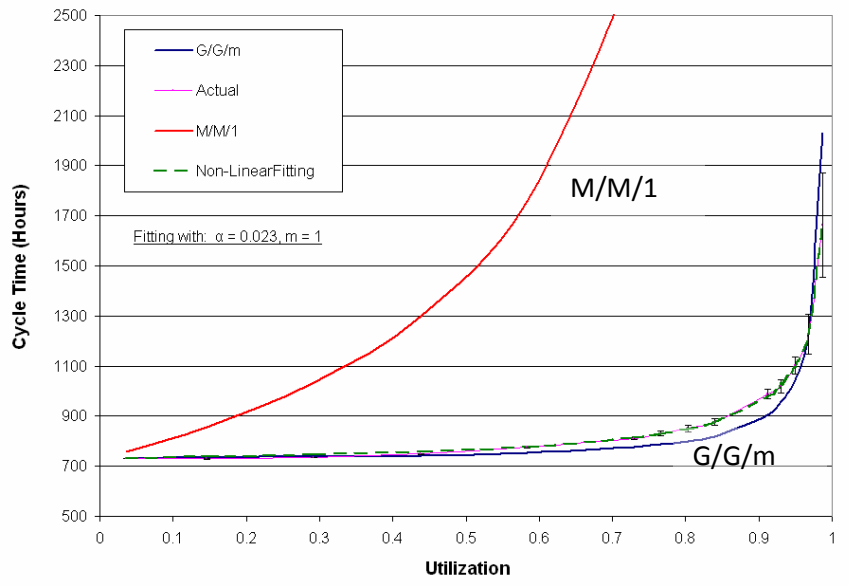


Figure 2. Initial FES results from black box approach

In Figure 2, utilization is estimated as (arrival rate)/(saturation arrival rate). While fitting the G/G/m model with m=1 appears to give a good fit, the errors actually are significant, reaching 75% at a utilization just over 90%. In contrast, the non-linear fitting model, which has been developed for this project and will be introduced in Section 6.2, is quite accurate, with maximum errors of about 10% over the range of utilizations examined.

The remaining technical work for the black box approach involves repeating the approach with several other simulations to establish a repeatable process and then developing computational methods to automate as much of the experimental and statistical work as is possible.

4.2) White Box Approach

In contrast to the black box approach, the white box approach assumes the simulation model itself can be examined in order to identify the specific structure and parameter values used. By "opening up" the simulation model, it is possible to create a "model of the model." This less detailed model should still retain the essential structure of the simulation model, by aggregating non-bottleneck processes in the simulation model, and focusing on bottleneck processes. With this less detailed model, the intent is to adapt existing analysis approaches, such as decomposition, to apply to the less detailed model.

There are two key technical issues in the white box approach:

(1) how to derive the less detailed model structure and parameters; and

(2) how to analyze the resulting model.

Although the white box approach requires more information than the black box approach, it is much less detailed than the original simulation models; in fact, it would be difficult to infer much about the original simulation structure from the associated white box model. The specific approach to constructing the white box model will be explained in Section 7.2.

Although a more detailed white box model is possible, in Phase I, the white box approach mainly focuses on the G/G/m based model. Compared with the black box approach, which is based on the G/G/1 model, the G/G/m based white box model needs to know the exact server numbers at the bottleneck and the 2nd bottleneck.

4.3) Variability Analysis

To answer the question of 'Risk' associated with the cycle time estimate, we need to model the cycle time variability by estimating cycle time quantiles along with the mean cycle times. To do this, we first save all Simulation experiment data and then fit the quantile curves. The procedure is explained in section 5.5) Modeling Cycle Time Variability.

Creating models for the 5% and 95% quantiles of cycle time is similar to creating models for the mean cycle time.

4.4) Introduction to Simulation Models Used

4.4.1) Doyle Center Model

DSN Innovations (or just DSN) is a "non-profit organization focused on bolstering U.S. manufacturing through research and innovations designed to improve manufacturing supplier network coordination, agility and efficiency." One of the tools they have developed is a flow shop simulator that can be quickly populated with data from a small manufacturer and used to evaluate WIP at each workstation and overall manufacturing cycle time. Although the kernel of the tool is ARENA®, it uses Excel as its data input interface, so users don't require working knowledge of ARENA®.

DSN provided a case based on a manufacturing system (so called Doyle Center Model) consisting of five workstations as shown in Figure 3. While stations 1 and 5 are visited only once, stations 2, 3 and 4 are visited multiple times. Stations 2, 3 and 4 can execute multiple job functions: station 2 can do two different recipes, station 3 can do three and station 4 can do seven. There is a total of 14 process steps. There is a constant 20-minute delay between the first and the second steps. Some steps may need to be reworked if the finished jobs are out of spec. All stations are composed of one single machine except for station 5, which contains 24 machines in parallel. The initial arrival process is Poisson. All dispatching rules are FCFS. The service time distribution is triangular and the data of each process step is shown in Table 2. Based on Table 2 and Figure 3, the capacities of stations 1 to 5 are 62.609, 51.429, 49.655, 49.021 and 54.857 jobs/day and service times are 23, 28, 29, 29.375 and 630 minutes, respectively. The fourth station is the bottleneck.

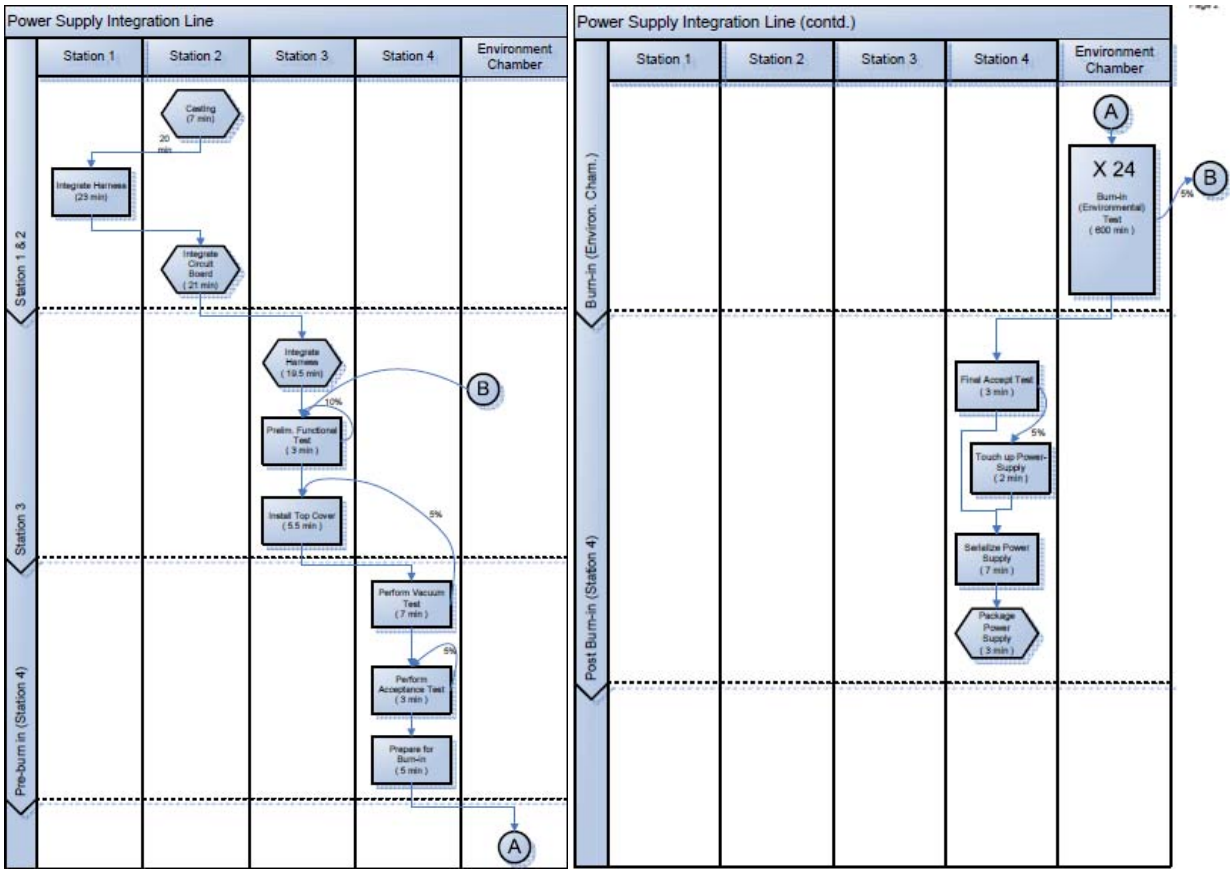


Figure 3. Process flow of the Doyle center model

Table 2. Process times of each process step

Process Number	Process Description	Process Resource	Processing Times (min.)		
			Min	Mode	Max
1	Bond Overlay and O-Ring	Station 2	6	7	8
2	Integrate Harness at Station 1	Station 1	22	23	24
3	Integrate Circuit Boards at Station 2	Station 2	20	21	22
4	Integrate Harness at Station 3	Station 3	18.5	19.5	20.5
5	Perform Preliminary Func Test	Station 3	2	3	4
6	Install Top Cover	Station 3	4.5	5.5	6.5
7	Perform Vacuum Test	Station 4	6	7	8
8	Perform Acceptance Test	Station 4	2	3	4
9	Prepare for Burn-in Test	Station 4	4	5	6
10	Burn-in (Environmental) Test	Environ. Chamber	600	600	600
11	Perform Final Accept Test	Station 4	2	3	4
12	Touch-up Power Supply	Station 4	1	2	3
13	Serialize Power Supply	Station 4	7	7	7
14	Package Power Supply	Station 4	3	3	3

4.4.2) Assembly Line Model – RC Module 1, 2 and 3

4.4.2.1) Introduction

For the purpose of studying the behavior of assembly lines, a three-stage assembly line model was provided by Rockwell Collins. Each of the three stages is modeled using ARENA® with a separate simulation file. The assembly line has rework within each stage and between the stages. The following diagrams represent the flow among components. The number of components for each type needed is given in the brackets after each component type.

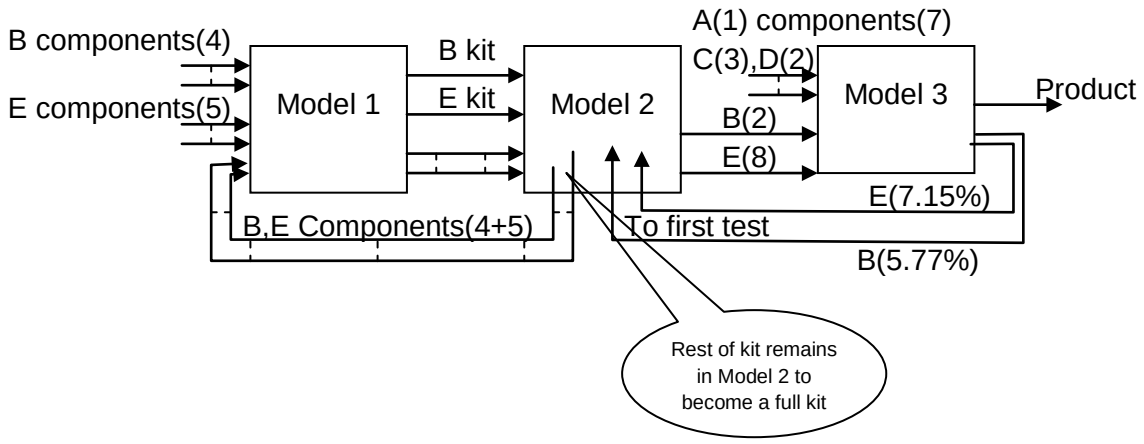


Figure 4. Graphical representation of the rework pattern

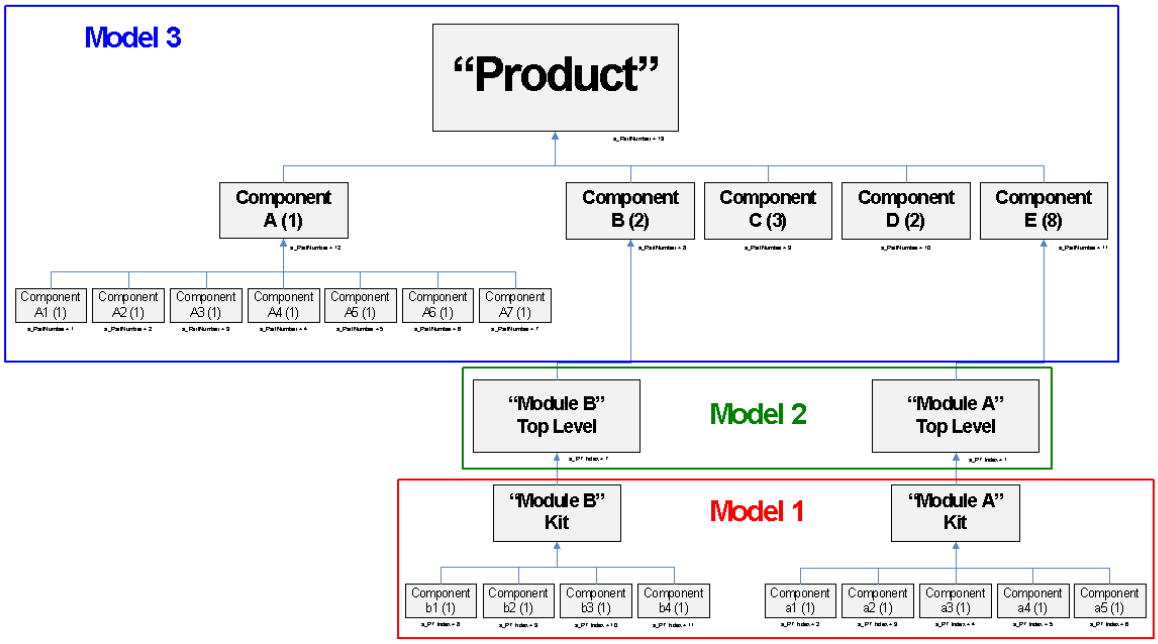


Figure 5. Product flow for all three models

4.4.2.2) Model 1

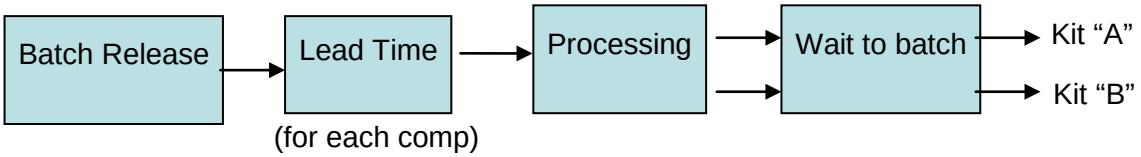


Figure 6. Block representation for Model 1

Model one has two outputs; one for Module “A” kit and one for Module “B” kit. Both these products share resources. This makes analysis of such a system difficult; as the cycle time, bottlenecks, and capacities for the two outputs are different and dependent on each other. The ratio of 8 Module “A” kits to 2 Module “B” kits is constant for all levels of the “Product” demand. This allows us to now have one input mix, but cycle times will remain separate.

The batch release occurs every three days and there are different lead times for each component. Each component follows a specific sequence through the stations and tests. Some stations have rework and there is a repair station where components may be routed. The tests are either manual or automatic.

Automatic Tests require the test technicians to be there only to start the test and check results. The Test Technicians are on a schedule 7:00 am to 3:30 pm while the machines run all the time.

Finally the components are routed to assembly where they wait till there is one of each of the components to form a kit and then they leave the system. Thus there is a wait to batch queueing time, as well.

4.4.2.3) Model 3

There are two assemblies in this model. Component A is assembled from seven components and then Component A, along with four other component types, is assembled to form the “Product.”

It is similar to the Model 1 and is again like a job shop but a large number of components pass through the same machines in the same sequence.

The resources are in sets of five and use a priority rule defined for each type of resource; e.g., for resource 1; choose resource 1a always if it is available; then 1b if available and so on. This will lead to 1a having the largest utilization and 1e having the least.

Some stations have a combination of two resources, resource 2 and another resource which is not used anywhere else. Thus these stations are dominated by resource 2 and behavior is not affected by the other resources.

One of the issues we found unique in this model was that there were resources shared with other product lines and they were modeled as “Offline Resources” with a capacity of 100 servers. These may pose a challenge in terms of identifying the effective capacity as there is an underlying assumption that these offline resources will never block the operations being modeled. It may be a reasonable assumption only if it is logical for the real factory. For our

calculations of the theoretical capacity we made the capacity of the “Offline Resources” very large.

4.4.2.4) Difficulty with RC Model 1,2,3

Difficulty in analyzing assembly lines (limitation of the Black Box approach)

There are some inherent difficulties with assembly lines which needed to be explored. As Figure 7 below shows, the cycle time of the entire assembly line is dependent on which feeder line has the longest cycle time for each specific product instance. Thus to get the correct average cycle time for the final product, we require to take the maximum of the actual instantaneous cycle time of the components coming from each feeder line. This needs to be done at every assembly point till the last assembly point. Typically, simulation models do not collect data at this level of detail.

Since the models are separate and cannot be simulated together, we will calculate the intermediate cycle time from model 1 (CT_{E1}) and then add it to the cycle time of its feeder line (CT_{E2}). This requires us to have intermediate cycle time values before the assembly point in model 3; that may not be available in the Black Box approach.

Hence we find that this difficulty to integrate the models would apply to any assembly line that does not have a single stochastically dominating feeder line. For example; in the diagram below, we find that CT_A is a part of the Model 3 and thus we need to take the maximum at A’s assembly point inside Model 3. Assuming that the simulation model has implemented the maximum, we still need to save the value of individual feeder lines if a WIP Profile needs to be estimated.

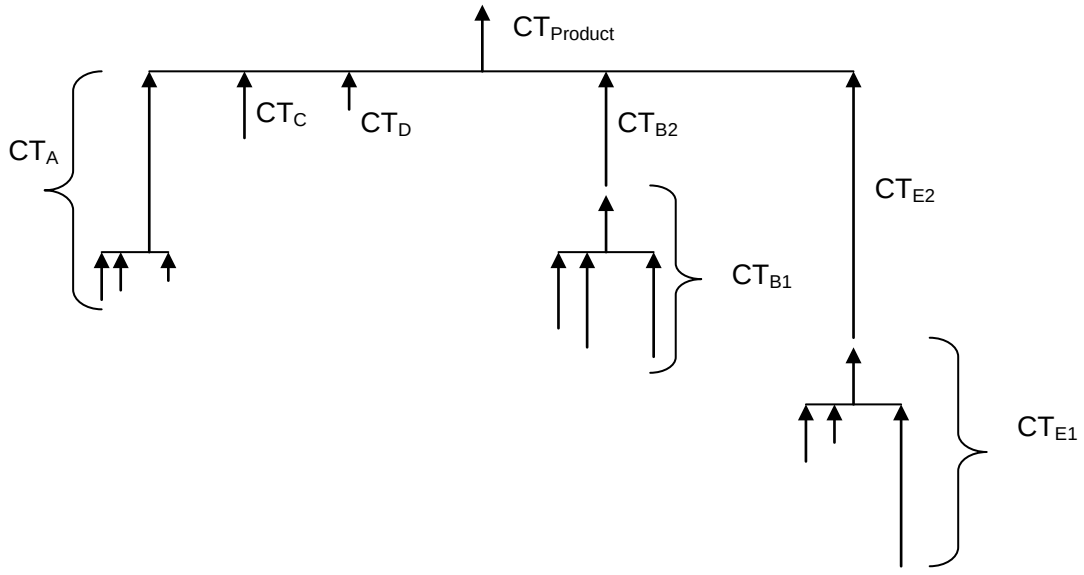


Figure 7. Graphical representation of cycle times

Also, CT_A needs to be recorded as we need to compare it to CT_{E2} (as well as the other feeder lines) to check which one is the feeder line with the longest cycle time. Then CT_A may need to be used to calculate the final cycle time. It may be possible that neither CT_A nor CT_{E2} is stochastically dominating and for some instances CT_A is larger and for some CT_{E2} is larger. Thus there are many difficulties with a black box approach on a single model which has assembly line structure. Special care needs to be taken to add additional maximum operators at all assembly points. Also WIP profile will not be available unless we go to the white box and share the assembly line structure.

Difficulty integrating the three RC models coming from a single factory

Estimating rework flows and obtaining actual cycle times under the existence of rework is a difficult task when the three stages are modeled separately. Obviously, this task could be avoided if the integrated model were available.

As the integrated model is not available, the following step-wise methodology was developed to produce cycle time estimation with the existence of rework. This procedure is required because the 3 stage model was created from a single factory; and it is unlikely that we would see this sort of intensive rework routes between two different real factories in a supply chain. Thus since this particular model results are not representative of a real supply chain; we have avoided the stepwise integration and will complete our analysis on the integrated model when made available.

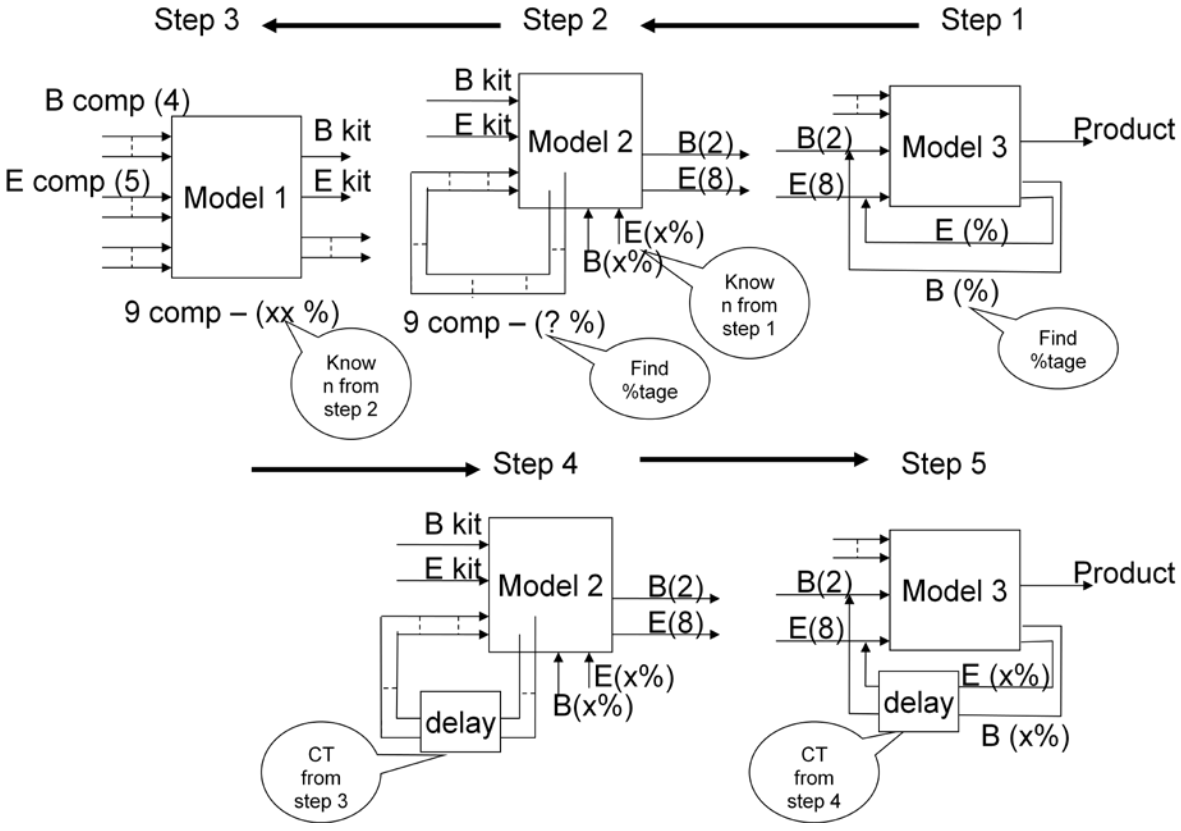


Figure 8. Graphical representation of the methodology proposed

5) Procedure for the FES approach

5.1) Step1: Determine the System Capacity and Warm-Up Period

5.1.1) Determining System Capacity

When we cannot see inside the simulation model but can only measure outputs for a given input, we need a methodology to find the capacity of the factory represented by the simulation model.

We use some estimated starting values as the arrival rate and measure the cycle time. If the simulation breaks down (i.e., gives error saying too many entities in the system), then we know that the capacity is less than this starting value; else we use a larger value. This methodology is like binary search; we know that with input rate at capacity (actually just a little less than that) the cycle time will keep increasing and may not stabilize; just above the capacity the system is sure to breakdown as the number of entities in the simulation will explode. Single Replication lengths in the range of 110 to 200 years are used for this purpose with warm-up period of about 100 years. The capacity figure generated by this method may be inaccurate by a small margin.

5.1.2) Determining the Warm-Up Period

For further analysis, it is important to know how long it takes for a simulation to reach steady state. For finding capacity, we made the warm up to be 100 years hoping it will be enough. Since we are going to do a lot of simulation replications, we will determine how long it takes to reach steady state and use a warm-up period that is greater. To do this, we use a very high utilization level as all lower levels will have smaller warm-up periods.

In this method we observe the overall Factory Work in Progress (WIP_f) in the system which can be easily obtained by saving the difference between the instantaneous Count-In and Count-Out values during simulation. Saving these values at daily time intervals and then finding the moving average over 50 (or 100) days and plotting with respect to time gives us the following graphs. When the moving average is found to be constant, we can assume that the system has reached steady state and using a warm up greater than this is enough. It is acceptable to overestimate the warm up, but care should be taken not to underestimate it.

For Model 3, the following three Work in Progress (WIP) graphs (given by instantaneous ‘Count In’ – ‘Count Out’) are for different arrival rates which give insight as how to estimate warm up. In the following case, we find that the WIP is stable and using about 50 years warm up would be appropriate.

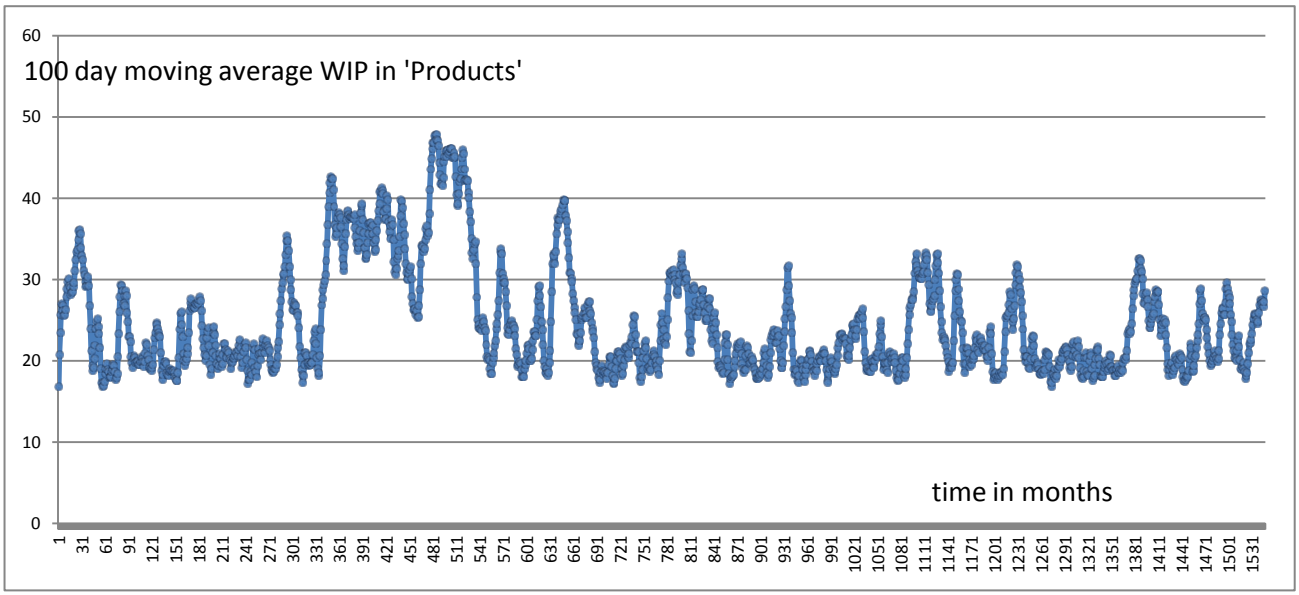


Figure 9. 100 days moving average for WIP when $\lambda = 448$

In Figure 10 we find that the moving average WIP stabilizes after around 120 years. Thus clearly the capacity of the system is at least 449. The same is seen if we use moving average of the cycle time instead of the WIP.

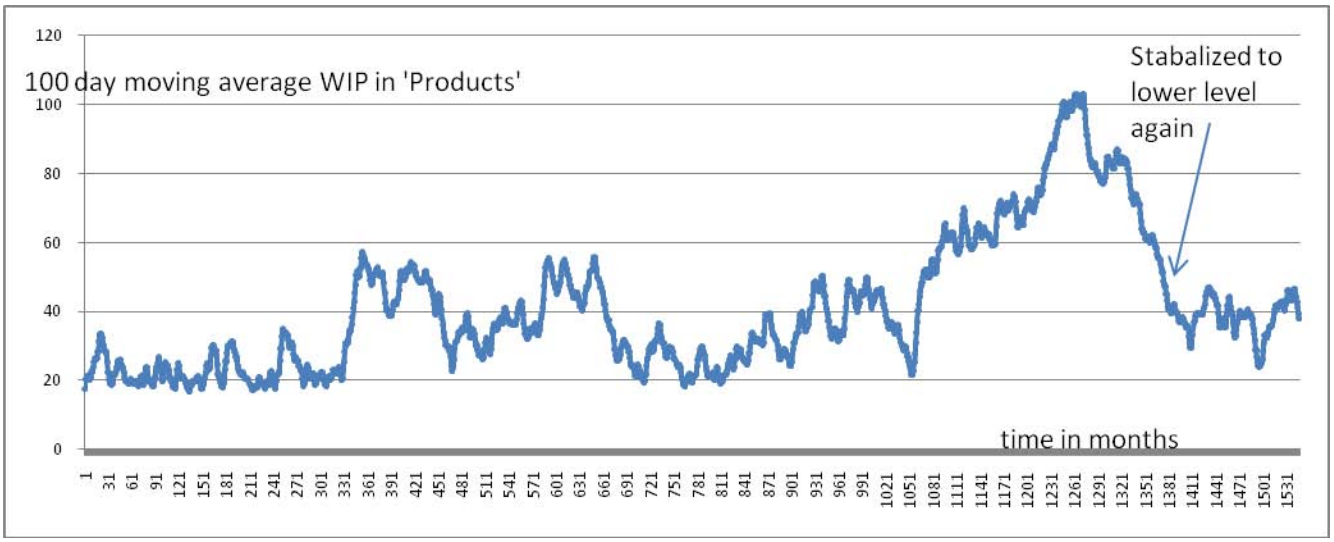


Figure 10. 100 days moving average for WIP when $\lambda = 449$

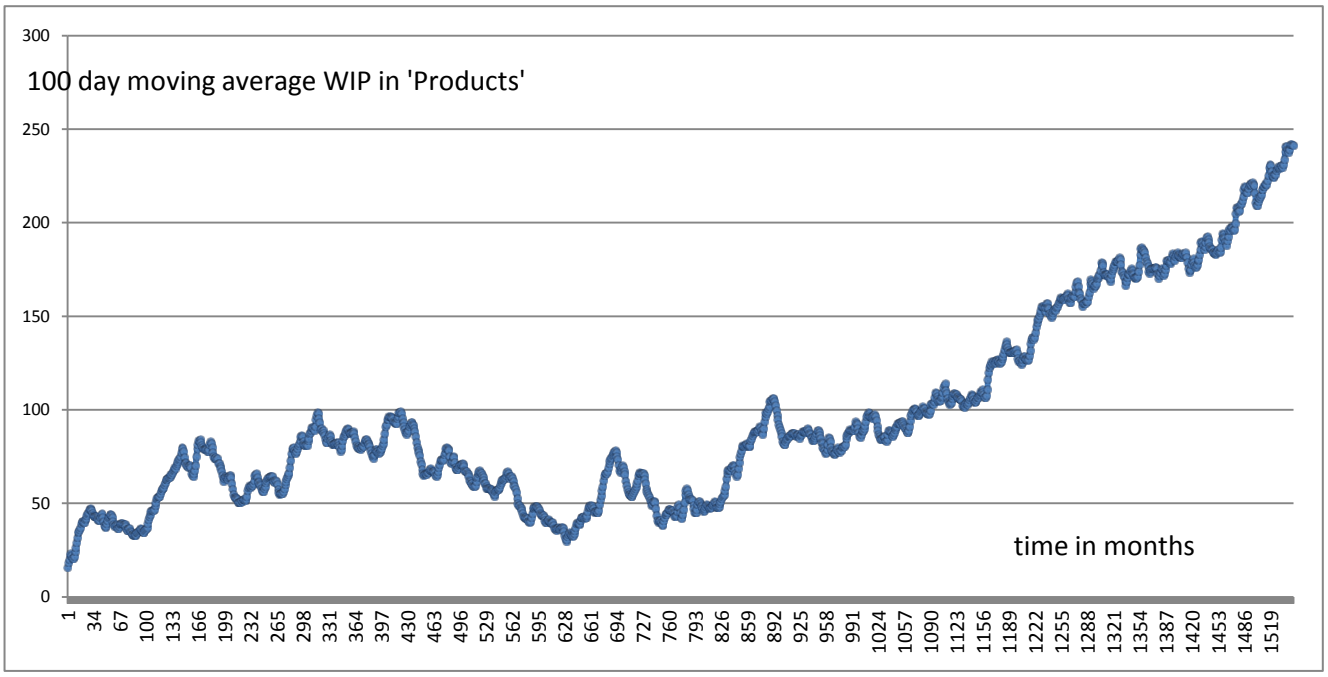


Figure 11. 100 days moving average for WIP when $\lambda = 450$

In Figure 11 we find that the WIP continues to increase. Thus the capacity of the system is 450 (actually it is between 449 and 450). Also, using warm up of 100 years can be considered appropriate as we saw that at $\lambda = 449$ the WIP started becoming unstable.

5.1.3) Determining Number of Servers in System Bottleneck (only for White Box approach)

For the white box approach we need to know the number of servers (m_{BN}) in the system bottleneck resource as well as the number of servers in the second system bottleneck. For this we must be able to see inside the model and first find which resource is the bottleneck and then find the number of servers it has. The same needs to be repeated for the second bottleneck as well to obtain (m_2).

This is where we run the simulation at a high utilization level to save the values of all the machine work-in-progress at average levels. The bottleneck resource should have much larger average work-in-progress than the other resources. If the second bottleneck has similar capacity (e.g., bottleneck =50 and second bottleneck=49) then it would be difficult to distinguish between the two. Also, there may be more than one resource with the same capacity. In that case, the machine that is first visited by jobs in the process flow is the bottleneck machine.

There is a complication when a workstation with two resources is the bottleneck. In this case, we suggest using the lower number of servers among the two resources as the number of servers (m_{BN}) in the bottleneck. For the second bottleneck we need to find next bottleneck workstation.

This complication might be avoided by fitting m and m_2. However, this leads to a computationally complex model, and some statistical software packages simply cannot perform the necessary nonlinear optimization. Consultation with experts in statistical analysis indicates that a C-code based solution specific to this problem might be developed based on the "Numerical Recipes"[8] implementation of the Levenberg–Marquardt algorithm[9] [10]. This is beyond the scope of this Phase 1 activity.

5.2) Step 2: Collect cycle times for a range of utilization values

From Step 1, the Capacity of the Factory is known. System utilization will simply be input rate divided by capacity. A good practice is to have 10 levels of utilization between 20% to 90% utilization.

Once the arrival rates are available, they need to be entered as input to the automation code (as shown in Appendix A) and run to get corresponding cycle time values. At each utilization level we recommend using about 30 replications and for each replication collecting about 2000 observations of cycle time after the warm-up period. The warm-up period for low utilizations can be smaller than what was calculated earlier to save time though using the same warm up will result in correct values. It may take very long to collect the data from the simulation.

If variability analysis is also required, then each observed value of the cycle time must also be saved into a database for analysis later; else only the mean for each replication is required. Thus if variability analysis is not required; then about 30 data entries (one of each replication) will be there for each arrival rate (or utilization) level; saving its mean cycle time as an entry in a .csv file.

5.3) Step 3: Determine the Factory Process Time (PTf)

To fit the model, we need to first calculate the Factory Process Time (PT_f) using simulation. We can input a batch of one product and measure the cycle time to estimate PT_f.

To estimate the Factory Process Time we may run simulation at low utilization levels and observe the lowest cycle time. The lowest cycle time observed may not be the true Factory Process time due to batching effects as explained in Appendix B. If simulation data is available for low utilizations (i.e., 0%-20%), then we can use this method for calculating PT_f which is the lowest mean cycle time of the low utilization levels; else it we can use the cycle time of batch size of one.

[8] *Numerical Recipes: The Art of Scientific Computing, Third Edition* (2007) is published in hardcover by Cambridge University Press (ISBN-10: 0521880688, or ISBN-13: 978-0521880688).

[9] K. Levenberg, "A method for the solution of certain problems in least squares, *Quart. Appl. Math.*, 1944, Vol. 2, pp. 164–168

[10] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J. Appl. Math.*, 1963, Vol. 11, pp. 431–441.

5.4) Step 4: Fit the model to the data collected

5.4.1) Black Box Approach

We use the model below to fit the data to generate the values for k_1, k_2 and k_3. The model derivation is provided in section 7.2.

$$CT \cong k_1\left(\frac{\rho_{BN}}{1-\rho_{BN}}\right)\frac{1}{\mu_{BN}} + k_2\left(\frac{\lambda}{k_3-\lambda}\right)\frac{1}{k_3} + PT_f \quad (1)$$

Where:

CT is the factory cycle time collected (.csv file)

λ is arrival rate collected (.csv file)

μ_{BN} is the capacity of the factory (calculated earlier)

ρ_{BN} is the utilization of the factory which is arrival rate by capacity of the factory (i.e., $\rho_{BN} = \lambda/\mu_{BN}$)

PT_f is the factory process time

k_1, k_2 and k_3 are the unknowns to be determined by regression analysis

5.4.2) White Box Approach

We use the model below to fit the data to generate the values for k_1, k_2 and k_3. The model is derived based on Section 7.2) Three-Parameter Model for Cycle Time Approximation and the results from Sakasegawa[11] (1977).

$$CT \cong k_1\left(\frac{\rho_{BN}^{\sqrt{2(m_{BN}+1)}-1}}{(1-\rho_{BN})}\right)\frac{1}{m_{BN}\mu_{BN}} + k_2\frac{\left(\frac{\lambda}{m_2k_3}\right)^{\sqrt{2(m_2+1)}-1}}{\left(1-\frac{\lambda}{m_2k_3}\right)}\frac{1}{m_2k_3} + PT_f \quad (2)$$

Where:

CT, λ, μ_{BN}, ρ_{BN}, PT_f, k_1, k_2 and k_3 are as in black box

m_{BN} is the number of servers of bottleneck resource

[11] Sakasegawa. H. 1977. An approximation Formula $L_q = \square\square^{\sqrt{\square}}/(1-\square)$. *Annual of the Institute for Statistical Mathematics*. 29 (A): 67–75.

m_2 is the number of servers in the second bottleneck resource

The code for fitting in R and other fitting issues can be seen in the Appendix B.

5.5) Modeling Cycle Time Variability

As mentioned earlier, if variability analysis is required then we must save instantaneous values at each utilization level for all 30 replications. This generally increases the simulation time drastically as very large amounts of data is collected and saved to files.

This data is then used to find the 5% and 95% quantile values for each utilization level. Thus we get a graph for 5% and 95% single data points (in contrast to 30 replication means in regular analysis).

Fit the 5% and 95% quantile data the same way as we do for mean cycle time as shown above, but recalculate PT_f using the 5% and 95% quantile data respectively.

6) Results

For the models described above in Sec 4.4, we found the cycle times for the Doyle Center Model, Models 1 and 3 of the RC Models. For Model 1 the cycle time for Product A (Product E when referenced with Model 3) was used only as it was found to be stochastically dominating feeder line in Model 3.

We used these models to simulate and fit the cycle time and obtain the residual standard error and maximum error percentage. The procedure used to obtain these results is detailed in Section 5) on page 17.

It should be noted that the results are very sensitive to the value used for the unknown Factory Process Time (PT_f). Therefore, it is important to have a good estimation of PT_f. The following sections show the residual standard errors and Maximum Error Percentage:

Table 3. Residual Standard Error for Simulated Process Time at Low Utilization Level

Residual standard error			
	Doyle Center	Model 1	Model 3
Black Box	0.006233	0.02645	0.2611
White Box	0.006233	0.01338	0.1647

Table 4. Maximum Error Percentage for Simulated Process Time at Low Utilization Level

Maximum Error Percentage			
	Doyle Center	Model 1	Model 3
Black Box	1.0794	0.2968	5.8007
White Box	1.0794	0.2077	3.3276

For Doyle Center Model the White Box is the same as the Black Box as the bottleneck and second bottleneck were found to be single servers.

The figures below are the fitted graphs for the results given by simulating for PT_f as shown in Table 3.

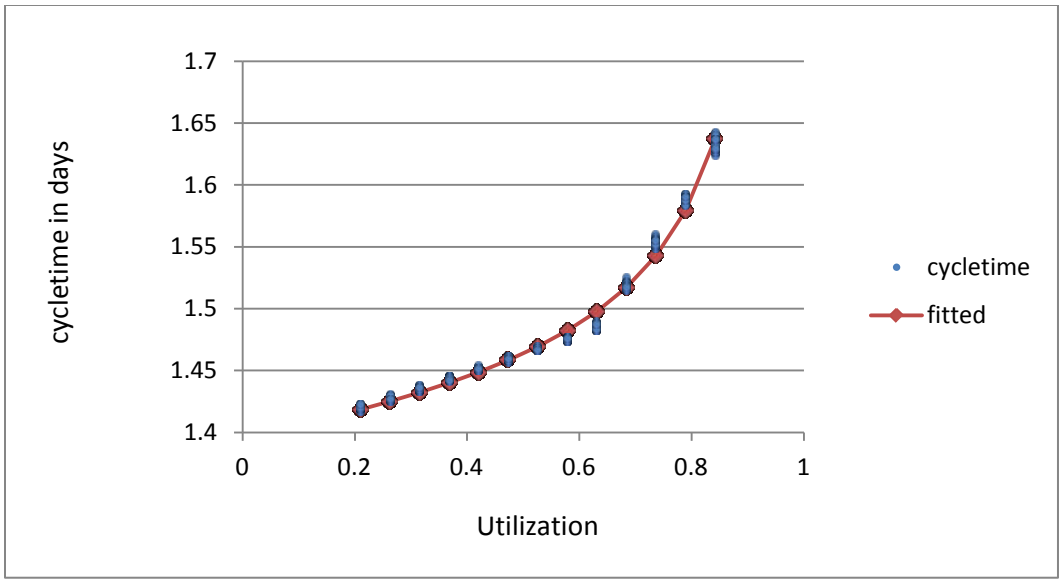


Figure 12. Doyle Center – Black Box and White Box

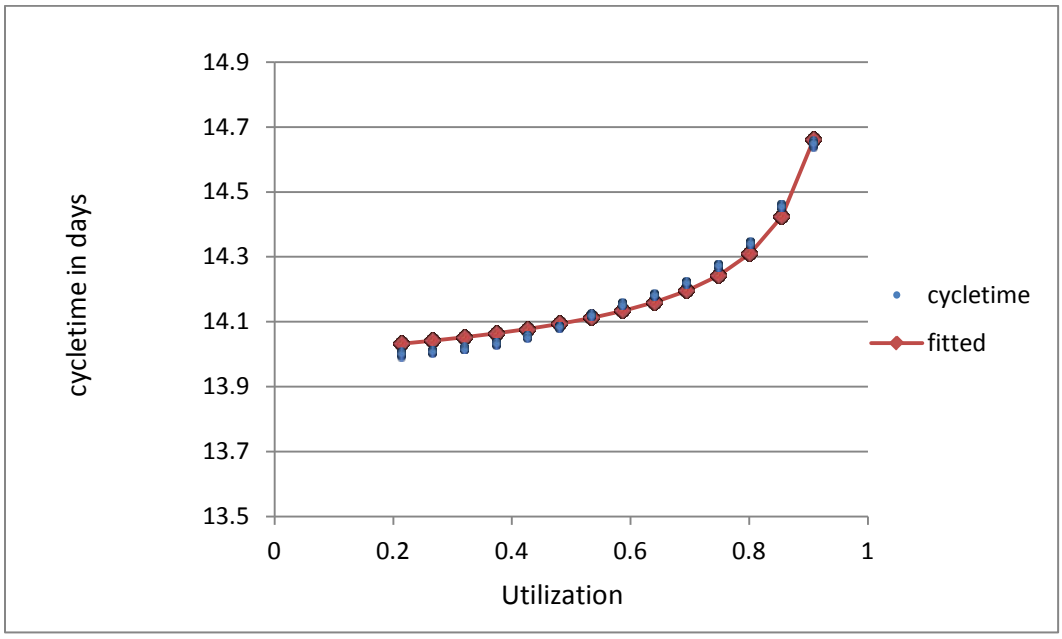


Figure 13. Model 1 – Black Box

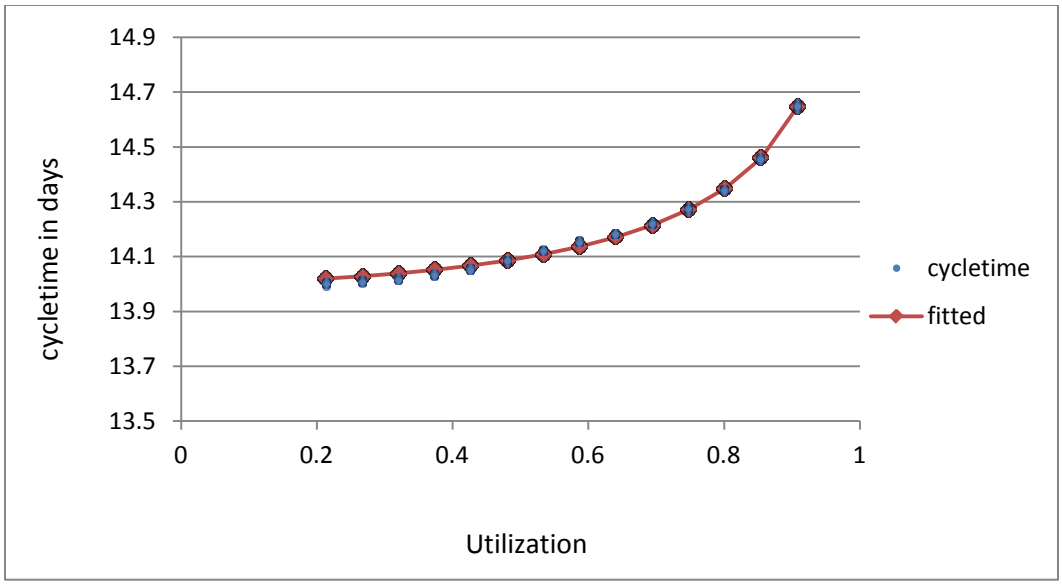


Figure 14. Model 1 – White Box

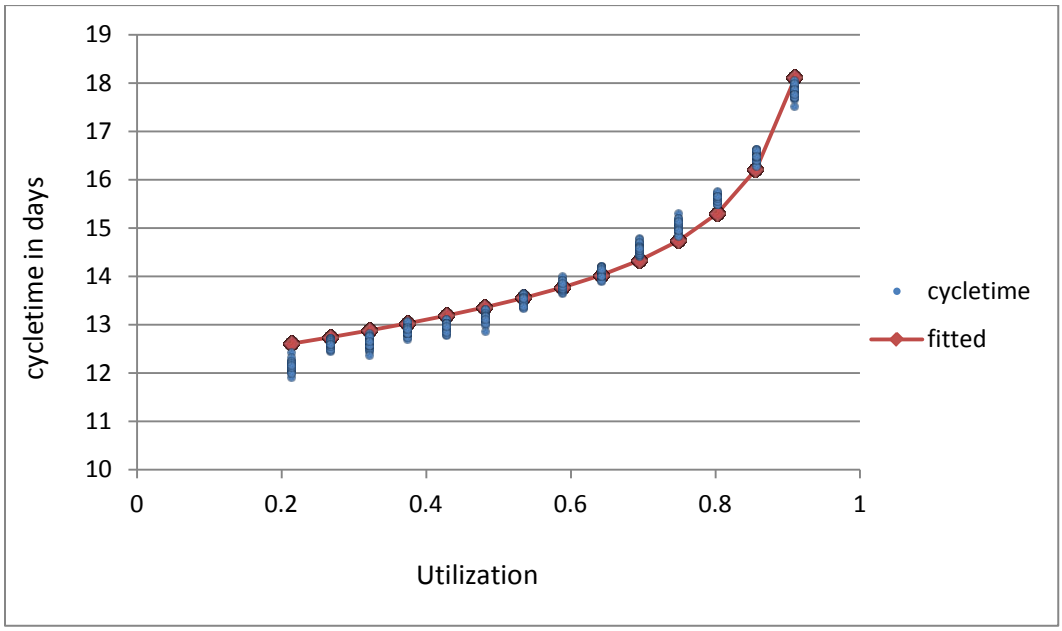


Figure 15. Model 3 – Black Box

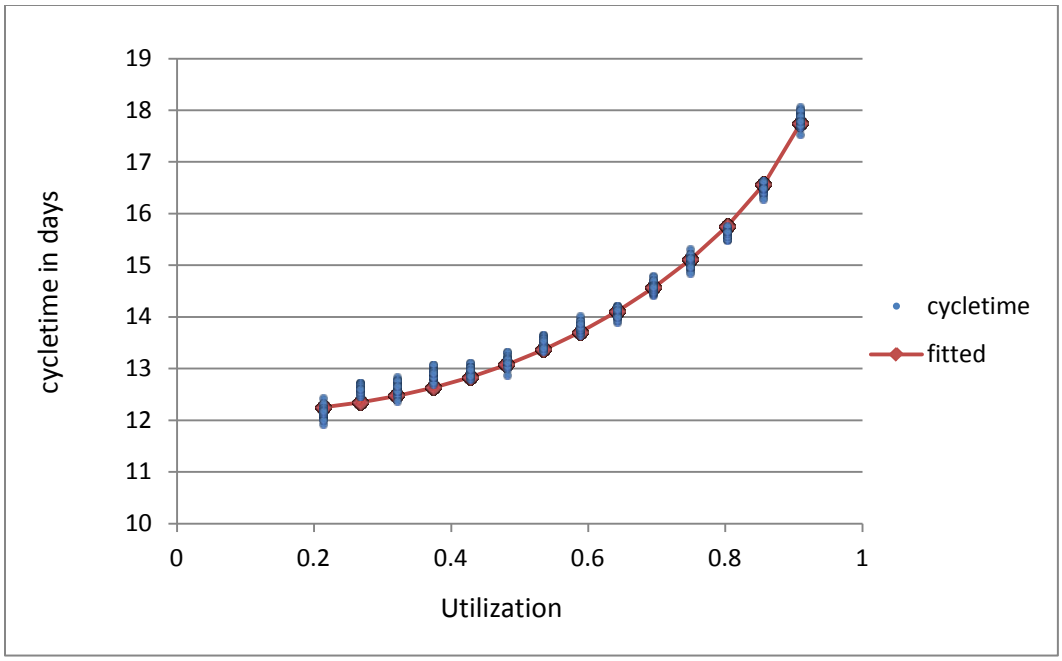


Figure 16. Model 3 – White Box

6.1) Result for Variability Analysis

The table below shows the Residual Standard Error and Maximum Error Percentage for the fits for the 95% and 05% quantiles.

Table 5. Results for Variability Analysis

	95% quantile	mean	05% quantile
Residual Standard Error	0.3086	0.1754	0.2008
Maximum Error Percentage	3.1213	2.9555	6.5770

We find that the quantile fit as well as the mean and this method can be used to model the prediction intervals using the data collected.

The graph below is the graph fitted for the quantile for Model 3.

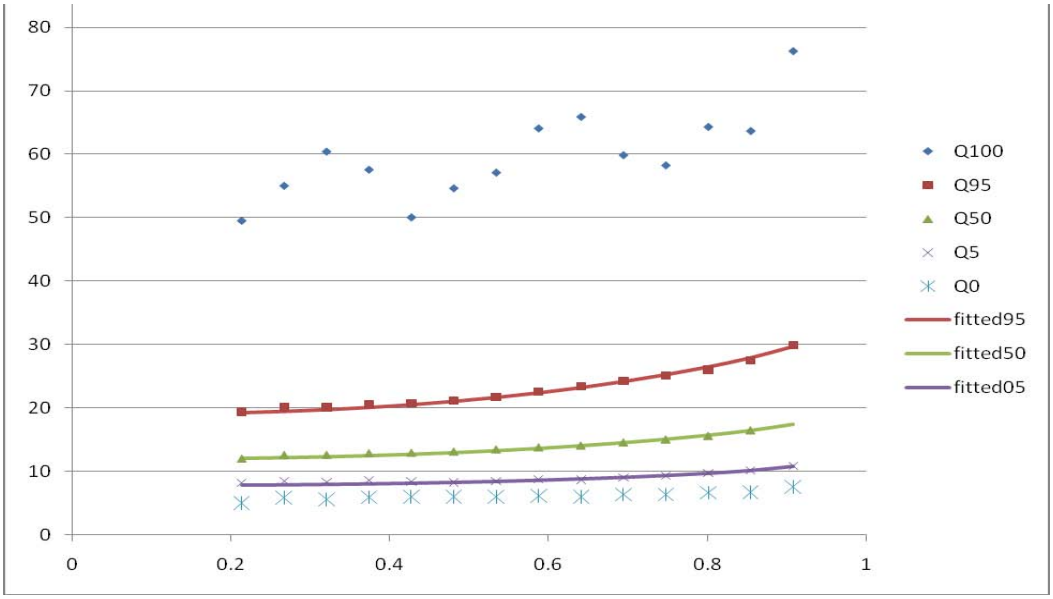


Figure 17. Model 3 - Quantile fitted

7) Technical Discussion

The previous discussion focuses on the operational procedures. In this section, we are going to give the theoretical background for the previously introduced procedures. Therefore, this section will be more theoretical. Readers who only want to understand the operational procedures may skip this part. However, it should be noted that the assumptions and limits of the proposed models can be explored only after the derivation of the models is fully understood. We will first start from the prior research and then develop the three parameter model for cycle time approximation.

7.1) Prior Research

There is significant archival literature on the analysis of networks of processes, much of it addressed to communication and computing networks. However, there is significant literature relevant to the analysis of manufacturing and logistics networks as well. Figure 18 below provides a roadmap to the relevant literature which falls into the category of "queuing network" analysis methods.

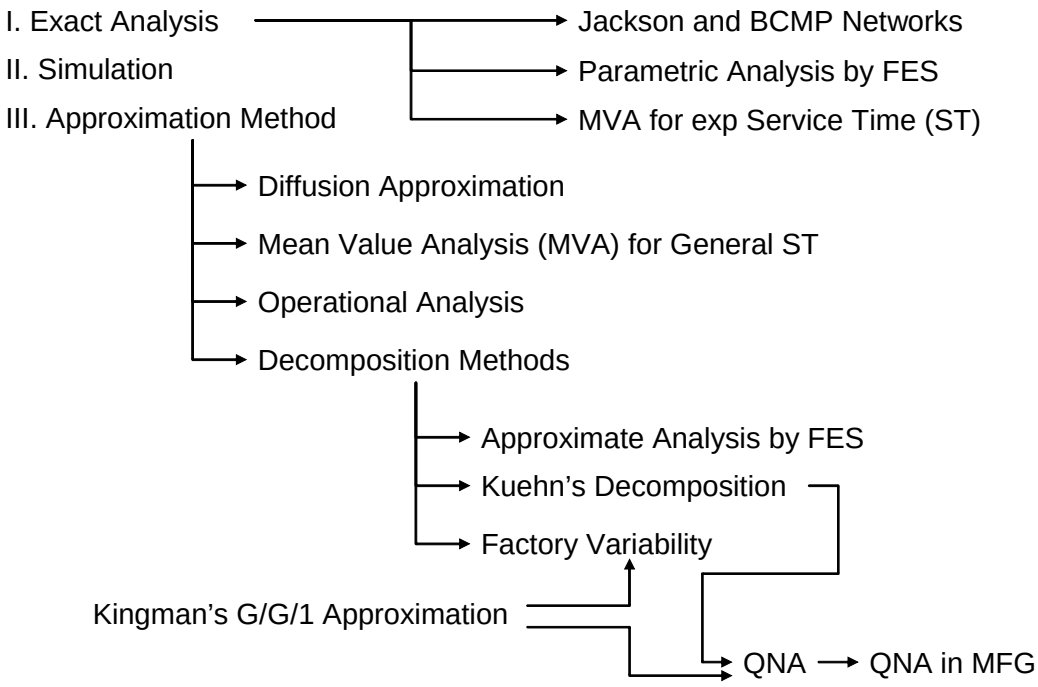


Figure 18. Roadmap for queuing network analysis methods

Exact methods of analysis require very stringent assumptions about the distributions of arrival and service times, the available queuing capacity in front of each server, and the processing

discipline for jobs waiting in queue. These assumptions preclude, for example, the kinds of service times or queuing capacity generally seen with highly automated processes.

Simulation methods, of course, permit high fidelity representation of the target system, but are expensive and time consuming to develop and analyze. The proposed research is an attempt to circumvent the time and cost associated with conventional simulation methods.

There is a large body of research on approximation methods. Diffusion approximations[12], e.g., allow less restrictive assumptions about arrival and service time distributions, but still assume unlimited queuing capacity and add the assumption that queues are never empty.

For systems that are assumed to be "closed," i.e., the number of jobs in the system remains constant, or equivalently, a new job is only introduced when an existing job is completed, Mean Value Analysis[13] (MVA) can be applied for general service times. While MVA still requires a number of restrictive assumptions, it has been used with some success to model flexible manufacturing systems[14].

Operational analysis[15] uses a state space balance equation, where state is defined as a vector of queue lengths, and requires conditional invariance assumptions. There is some concern about the computational viability of operational analysis, although Dallery[16] was able to develop some approximation methods by making additional assumptions about service time and job routes.

There are a number of decomposition approaches. Chandy[17,18] first introduced the "flow equivalent server" approach for analyzing a queuing network. The most widely known decomposition method is QNA[19,20], which has been extended to incorporate machine breakdown,

---

[12] Reiser, M. and H. Kobayashi (1974). "Accuracy of the Diffusion Approximation for Some Queueing Systems." IBM Journal of Research and Development **18**(2): 110.

[13] Reiser, M. and S. S. Lavenberg (1980). "Mean-Value Analysis of Closed Multichain Queuing Networks." J. ACM **27**(2): 313-322.

[14] Suri, R. and R. R. Hildebrant (1984). "Modelling Flexible Manufacturing Systems Using Mean-Value Analysis." Journal of Manufacturing Systems **3**(1): 27.

[15] Denning, P. J. and J. P. Buzen (1978). "The Operational Analysis of Queueing Network Models." ACM Comput. Surv. **10**(3): 225-261.

[16] Dallery, Y. and X.-R. Cao (1992). "Operational Analysis of Stochastic Closed Queueing Networks." Performance Evaluation **14**(1): 43-61.

[17] Chandy, K. M., U. Herzog and L. Woo (1975). "Parametric Analysis of Queuing Networks." IBM Journal of Research and Development **19**(1): 36.

[18] Chandy, K. M., U. Herzog and L. Woo (1975). "Approximate Analysis of General Queueing Networks." IBM Journal of Research and Development **19**(1): 43

[19] Whitt, W. (1983). "The Queueing Network Analyzer." The Bell System Technical Journal **62**(9): 2779.

[20] Segal, M. and W. Whitt (1989). "A Queueing Network Analyzer for Manufacturing." Teletraffic Science for New Cost-Effective Systems **Networks and Services**(ITC-12): 1146.

batch service, changing lot sizes, repair, and fractional yield. Generally, the decomposition approach is used to estimate average queue length and waiting time. Wu[21] uses a decomposition approach to try to understand total factory variability, considering utilization and throughput bottlenecks.

7.2) Three-Parameter Model for Cycle Time Approximation

A factory may have long process flow sequences with reentry and rework, where each workstation may be composed of multiple servers with different capabilities and both random queueing time and asynchronous queueing time may exist. Complex dispatching rules other than FCFS may be applied to each workstation. Under these conditions, understanding the behavior of a factory may not be an easy task.

However, if we want to optimize factory performance, describing the behavior of a factory quantitatively is essential. Rather than analyzing all activities in detail, we derive an approximate model by capturing the main underlying structure of a factory. Based on Wu (2009)[22], the model is as follows,

$$\begin{aligned} CT &= \alpha_{BN}\left(\frac{\rho_{BN}}{1-\rho_{BN}}\right)\frac{1}{\mu_{BN}} + \sum_{i\neq BN} f_i\alpha_i\left(\frac{\rho_i}{1-\rho_i}\right)\frac{1}{\mu_i} + PT_f \\ &\cong k_1\left(\frac{\rho_{BN}}{1-\rho_{BN}}\right)\frac{1}{\mu_{BN}} + (n-1)k_2^{'}\left(\frac{\lambda/k_3}{1-\lambda/k_3}\right)\frac{1}{k_3} + PT_f \\ &= k_1\left(\frac{\rho_{BN}}{1-\rho_{BN}}\right)\frac{1}{\mu_{BN}} + k_2\left(\frac{\lambda}{k_3-\lambda}\right)\frac{1}{k_3} + PT_f, \end{aligned} \tag{3}$$

where f_i is the contribution factor and can be approximated by a function of $(\lambda, c_{a1}^2, ST_i, c_{ei}^2)$ for i = 1 to n. PT_f is total processing time, which is the minimum time that a job needs to complete its process. In this model, the first term can be interpreted as corresponding to the bottleneck queuing time with k_1 as the bottleneck variability. The second term can be interpreted as corresponding to queuing time at a composite non-bottleneck station, with the constant k_2 approximating the variability of this composite station (representing the $(n-1)$ non-bottleneck stations), and k_3 representing the composite non-bottleneck capacity.

When there is reentry or rework, capacity is the reciprocal of the summation of all service times weighted by the rework rate. Therefore,

$$1/\mu_i = \sum_{j=1}^{l} w_j \times ST_j,$$

[21] Wu, K. (2005). "An Examination of Variability and Its Basic Properties for a Factory." Semiconductor Manufacturing, IEEE Transactions on **18**(1): 214-221.

[22] Wu, K. (2009). "An New Results in Factory Physics." Ph.D Thesis, Georgia Tech.

where l is the total reentry and rework frequency at station i, w_j is the rework rate when ST_j is the length of a rework (and w_j is 1 when it is the length of a reentry). Since there are three parameters in Eq. (3), we call it the 3-parameter model.

Although Eq. (3) is motivated by the underlying structure of tandem queues, as we will see later, it performs very well for the practical manufacturing systems examined, even with reentry and rework. When applying Eq. (3) to a specific factory, the values of k_1, k_2 and k_3 should be determined considering practical issues such as reentry, dispatching rules and interruptions. Obviously, calculating k_1, k_2 and k_3 analytically is difficult. One way to determine their values is by multiple regression analysis, if the historical performance curve is available. Then, factory variability can be approximated by k_1 and k_2. *We say the performance of a factory is improved, if the value of k_1 or k_2 becomes smaller at a given traffic intensity*. The parameters k_1 and k_2 describe the variability of a factory in the approximate model of Eq. (3). Therefore, considering both Eq. (3), it may be concluded that factory variability can be lowered by reducing the service time variability, the initial arrival process variability, or the number of non-bottlenecks.

If there are multiple bottlenecks (i.e. more than one server, which has the same highest utilization), only the one with the smallest sequence number is marked as the bottleneck. Eq. (3) gauges the variability of a manufacturing system from the viewpoint of the bottleneck, but adding a correction term to consider the impact from non-bottlenecks.

8) Neutral Data Format Definition

As mentioned by the National Academy report on "Modeling and Simulation in Manufacturing and Defense Acquisition" discussed in the introduction, a "standardized external model representation" is needed. A standard format to represent the FES can be used for several purposes. As a neutral exchange format between two simulation models, where a sub-model is approximated as an FES, and then composed into another simulation model. It can also be used as the basis of an external model repository as described in the next section.

The models need to follow some conventions so they are compatible. In particular, the models need to be flow-compatible in that the types of objects that one outputs and the other inputs are the same. This is one reason why manufacturing and supply chain simulations are often difficult to make interoperate because, in manufacturing, simulations the objects are often individual products and in supply chain simulations the objects are usually shipments composed of multiple products. In this example, batching and un-batching is needed at the interface to make the corresponding objects the same type.

The inputs and outputs for a single FES do not have to be the same type, for instance parts, may be inputs and assemblies can be outputs. In addition, an FES can be multiple inputs and outputs, for instance, different types of parts and assemblies. However, in the theoretical and experimental work performed in Phase I, we did not address the multiple input/output scenario.

If the models are developed using different simulation tools, the exchange format needs to be translatable from the format used by one tool to the format used by the other. The translation can be direct between the forms or through a third neutral format. Having a neutral exchange format in between offers some benefits in terms of input or output translators. If the number of simulation tools is N, and any tool can be a source or destination of the FES, the neutral format only requires 2N translators, while the direct approach requires N(N-1). In addition, if a tool changes its interface or format, then in the neutral approach only two translators need to change, while the direct approach 2N translators are affected by the change and need to be maintained.

Even if the simulation models are developed using the same tool, the model developer may want a neutral format for long-term retention. If a vendor changes their modeling software in some fundamental way between versions so that legacy models are no longer compatible, the legacy models will need to be remodeled to be reused. The neutral form ensures that legacy models will be upwardly compatible.

For exchange, the neutral format for the FES needs to contain the types or identities of the objects it inputs and outputs so that an FES can be composed into other simulations. Item Unique IDentification (IUID) is an asset identification system instituted by the DoD to uniquely identify a discrete tangible item or asset and distinguish it from other like and/or unlike tangible items.[23] It can also be used to identify batches or lots of items. We propose to use IUID to identify the server inputs and outputs and represent IUID with the compatible PLCS standard that we will discuss in the next section.

For the server relationship, we need to represent the functional relationship between the server rate and number of jobs in the queue. In the work by Georgia Tech presented in this report, this relationship is a mathematical equation. The equation has to be represented semantically so that it can be unambiguously

[23] http://www.acq.osd.mil/dpap/pdi/uid/index.html

translated into the machine readable form that each simulation package uses for representing mathematical equations. Content MathML, OpenMath[24], and PLIB Expressions in STEP are languages developed to semantically represent mathematical equations. There are also standard languages to represent equations used such as Open Office XML for Microsoft Office Excel and in the Open Document Format used by Open Office. However, software tool support for these languages is not widespread.

There is a web-based MathML generator that can create Content MathML representations translated from the syntax used by Mathematica.[25] As an example, below is the Content MathML for the server functional relationship k1 * (u/(1-u))*ST + k2 * (x/(k3 - x)) * ST + PT.

```
<math xmlns='http://www.w3.org/1998/Math/MathML'>
 <apply>
  <plus/>
  <ci>PT</ci>
  <apply>
   <times/>
   <ci>k1</ci>
   <ci>ST</ci>
   <ci>u</ci>
   <apply>
    <power/>
    <apply>
     <plus/>
     <cn type='integer'>1</cn>
     <apply>
      <times/>
      <cn type='integer'>-1</cn>
      <ci>u</ci>
     </apply>
    </apply>
    <cn type='integer'>-1</cn>
   </apply>
  </apply>
  <apply>
   <times/>
   <ci>k2</ci>
   <ci>ST</ci>
   <ci>x</ci>
   <apply>
    <power/>
    <apply>
     <plus/>
     <ci>k3</ci>
     <apply>
      <times/>
      <cn type='integer'>-1</cn>
      <ci>x</ci>
```

[24] www.openmath.org

[25] http://www.mathmlcentral.com/Tools/ToMathML.jsp

```
    </apply>
   </apply>
   <cn type='integer'>-1</cn>
  </apply>
 </apply>
</apply>
</math>
```

So we propose to use this to generate the neutral form for the server relationship that we will use in Phase 2 to translate to the simulation software specific representations for equations.

9) Model Repository Requirements

As mentioned by the National Academy report on "Modeling and Simulation in Manufacturing and Defense Acquisition" discussed in the introduction, modeling languages that can "support the structuring of large, complex models and the process of model evolution" are needed. In the last section, we described how a neutral format for the FES must contain identification of the inputs and outputs and the functional relationship for the server. In this section, we impose additional requirements on the neutral format so it can support a model repository to contain FESs to construct complex models and configuration manage the FES models as they evolve. These requirements consist of the meta-data needed to manage the models.

Fortunately, the same format that supports the IUID discussed in the previous section PLCS (Product Life Cycle Support), also known as STEP AP239 and its implementation in OASIS DEX, contains model configuration management and repository capabilities.[26] In particular, PLCS provides the model management capabilities needed for the FES repository by providing pre-built application modules and relationships between concepts such as

- Classification,
- Configuration effectivity,
- Product version,
- Organization (Structure and Type),
- Information rights,
- Location,
- Product data management,
- Approval, and
- Security classification.

Coincidentally, the recently (April 17, 2009) issued Air Force Instruction 63-101 states on page 120 that mandates that the Program Manager shall require the use of PLCS.

> "3.91.1.1. The PM shall require the use of International Standards Organization (ISO) 10303, *Standard for Exchange of Product (STEP*) *Model Data*, AP239, *Product Life Cycle Support*, for engineering data. "

[26] www.plcs-resources.org

A link to this document is below.

http://www.e-publishing.af.mil/shared/media/epubs/AFI63-101.pdf

This Air Force mandate for the use of PLCS is significant because if the FES uses PLCS, then there is a mechanism for contracting for the delivery of the FES models in a standard format. Other DoD services and agencies are likely to follow this best practice since the Aerospace Industries Association has recommended that DoD and contractors adopt PLCS for engineering data exchange.

http://www.aia-aerospace.org/pdf/wp_engineering-data-interoperability.pdf

In addition, model repositories and tools for developing exchange adaptors based on PLCS are becoming common. CostVision evaluated the Share-a-Space software and PLCS toolboxes from Eurostep and found them to be capable of meeting the model management and exchange requirements for the FES.

10) Future Work

Although we have successfully developed a reliable approximate model for system cycle time, there are still questions remaining to be resolved.

The current models have only considered the single product scenario. In practical manufacturing systems, a production line may deal with multiple products and the cycle time of each product may need to be estimated individually. In Phase I, we have developed the cycle time approximate model for single product scenario. The cycle time approximate models for multiple products are left for Phase II.

For an assembly line, in Phase I, we mainly focus on the total cycle time instead of the WIP profile. In order to compute the WIP profile for an assembly line, we have to know the detailed cycle time in each part of an assembly line. The data requirement and analysis is more than the White Box approach we have defined in Phase I. The more detailed WIP profile analysis is left for Phase II.

The current models developed in Phase I mainly focus on the impact from randomness effect. Although the impact from parallel batching has been considered into the model, the thorough discussion on the synchronization effects (such as shift schedule, dispatching rules and serial batching) is not done. The further investigation is left for Phase II.

Appendices

Appendix A

The code below is for RC Model 3 where we are using Lambda values in multiples of 25. The same can be easily edited according to requirements for any model.

```
% the automation function to run
Sub AutoRun()
    Dim noofarrivals As Integer
        noofarrivals = 19
    Dim arrivalrate(20)
    For i = 1 To noofarrivals
    arrivalrate(i) = i * 25
    Next i
    Dim s As SIMAN
    Set s = ThisDocument.Model.SIMAN
    For i = 1 To noofarrivals
        Model.Modules(Model.Modules.Find(smFindTag,
"object.13340")).Data("Initial Value(1)") = arrivalrate(i)
        Model.Go
        Model.End
    Next i
End Sub

% Saving the data after every replication
Sub ModelLogic_RunEndReplication()
Open ".\output.csv" For Append As #1
Dim s As SIMAN
Dim u As Double
Set s = ThisDocument.Model.SIMAN
u = s.VariableValue(s.SymbolNumber("v_Annual_Demand"), 0, 0)
Write #1, u, s.VariableValue(s.SymbolNumber("CT_Product1"), 0, 0),
s.VariableValue(s.SymbolNumber("CT_Product_SD1"), 0, 0),
s.VariableValue(s.SymbolNumber("CT_Product_CI1"), 0, 0),
s.VariableValue(s.SymbolNumber("CountB"), 0, 0),
s.VariableValue(s.SymbolNumber("CountE"), 0, 0),
s.VariableValue(s.SymbolNumber("CountP"), 0, 0)
Close #1
End Sub

% saving instantaneous values - vba block 1 is at the output point
(only for variability analysis)
Private Sub VBA_Block_1_Fire()
Open ".\run_output.csv" For Append As #1
Dim s As SIMAN
Dim u As Double
Set s = ThisDocument.Model.SIMAN
u = s.VariableValue(s.SymbolNumber("v_Annual_Demand"), 0, 0)
Write #1, u, s.RunCurrentReplication,
s.VariableValue(s.SymbolNumber("CT"), 0, 0)
Close #1
End Sub
```

Appendix B

The code below is R code for fitting the data when we know m and m_2 for the RC Model 3 case.

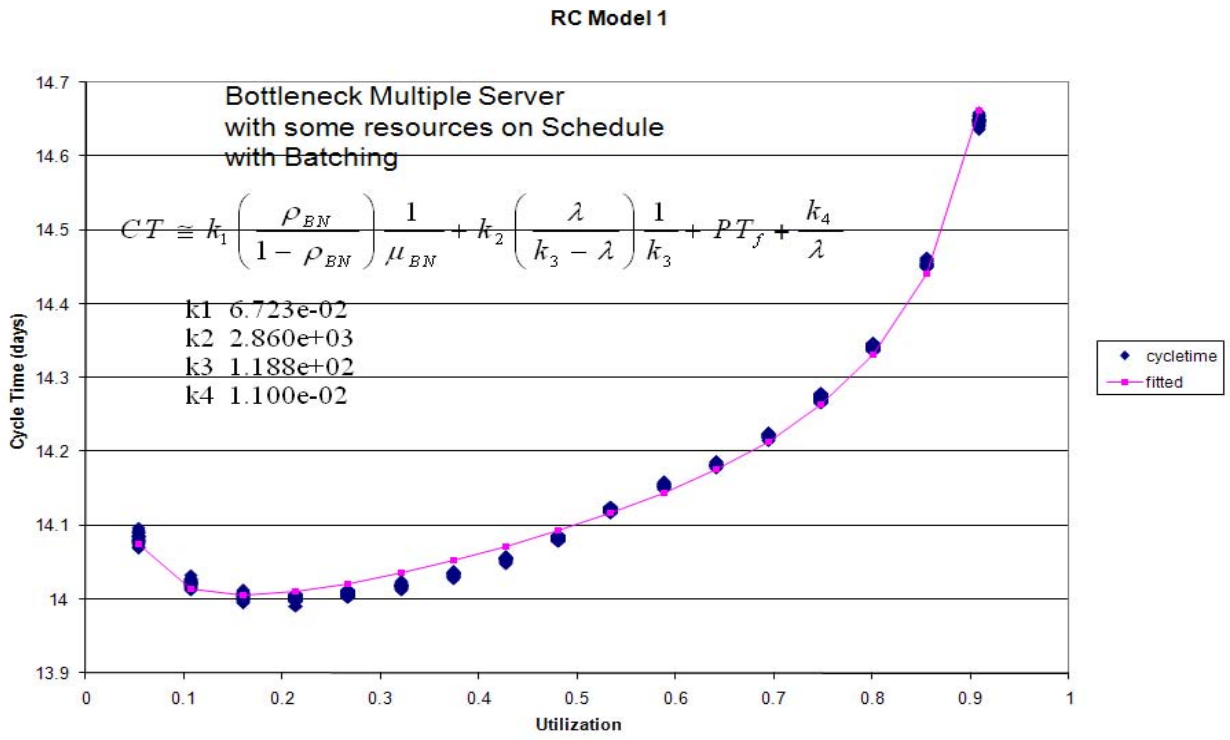
```
# clear all
rm(list=ls(all=TRUE))

# get the data from csv file
data1 <- read.csv('./model 3 with 30 reps.csv',header=T)

# input the constants
BN <- 467/360
m <- 5
m2 <- 5
PT <- 12.12790525

# get the starting values
model1.mod <- nls2(cycle time ~ k1 * (u^((2 * (m + 1))^0.5 - 1)/(m * (1 -
u)* BN)) + k2/(m2*k3) * ((u*BN/(m2*k3))^((2 * (m2 + 1))^0.5 - 1)/(1 -
(u*BN/(m2*k3)))) + PT,
data1,
control=list(maxiter=5000),
start=expand.grid(k1 = seq(.1,2, len = 3), k2 = seq(1, 1000, len = 3), k3
= seq(1, 1000, len = 3)),
algorithm="brute-force",
lower=list(k1=0,k2=0,k3=0),
trace=T)
coef(model1.mod)

# actual fitting is done using iterative method
model2.mod <- nls2(cycle time ~ k1 * (u^((2 * (m + 1))^0.5 - 1)/(m * (1 -
u)* BN)) + k2/(m2*k3) * ((u*BN/(m2*k3))^((2 * (m2 + 1))^0.5 - 1)/(1 -
(u*BN/(m2*k3)))) + PT,
data1,
control=list(maxiter=50000,tol=1e-15,minFactor=1e-5,warnOnly=T),
start=coef(model1.mod),
algorithm="port",
lower=list(k1=0,k2=0,k3=0),
trace=T)
summary (model2.mod)

# get the plot to see the fit
plot(data1$u,data1$cycletime)
lines(data1$u, fitted.values(model2.mod), type="b")

# calculate the maximum error percentage
ans <- 100*max(abs((data1$cycletime-
fitted.values(model2.mod)))/data1$cycletime)
ans

# return the fitted values to the .csv file
data1$fitted <- fitted.values(model2.mod)
write.csv(data1 , file = "./model 3 with 30 reps.csv")
```

Batching Effect and k4/λ term

We have observed that when there is batched input to a model, there are significant estimation errors at low utilizations. To obtain a better fit at low utilization when there is batching we added an extra term to the model, k4/λ term. The rationale for this term is that the average wait-in-batch time can be estimated by ((n+1)/2* 1/λ) where n is the batch size and λ is the batch arrival rate.

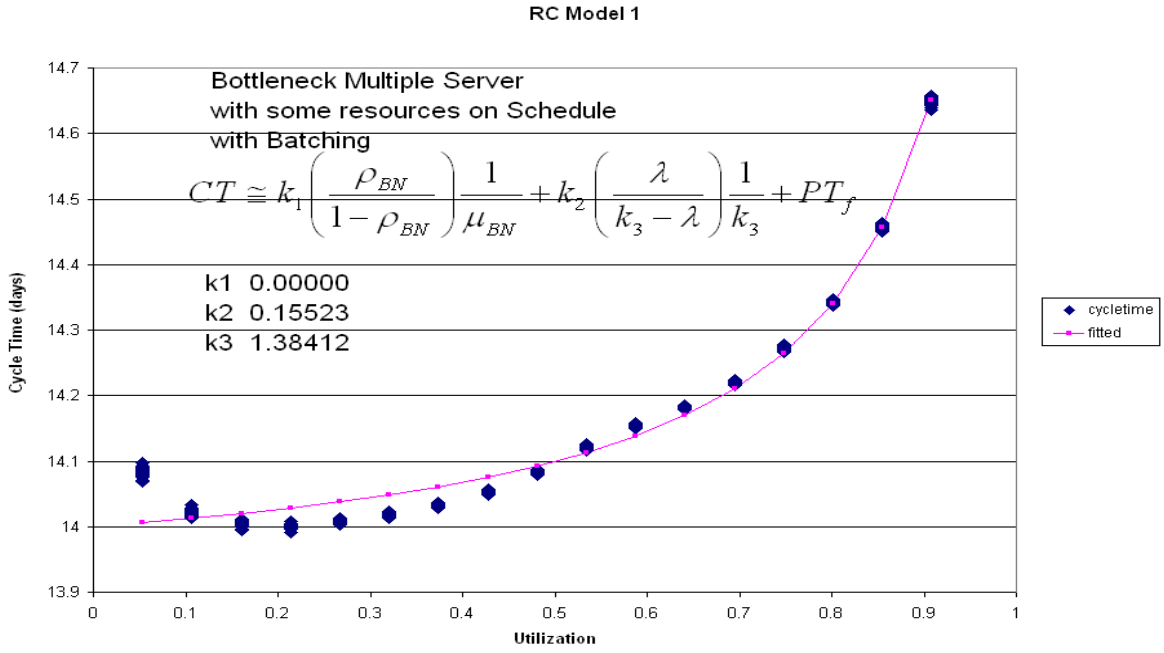
1.1) Model 1

The results for G/G/1 based approaches for model 1 up to 90% utilization with the k4/λ term are as follows:

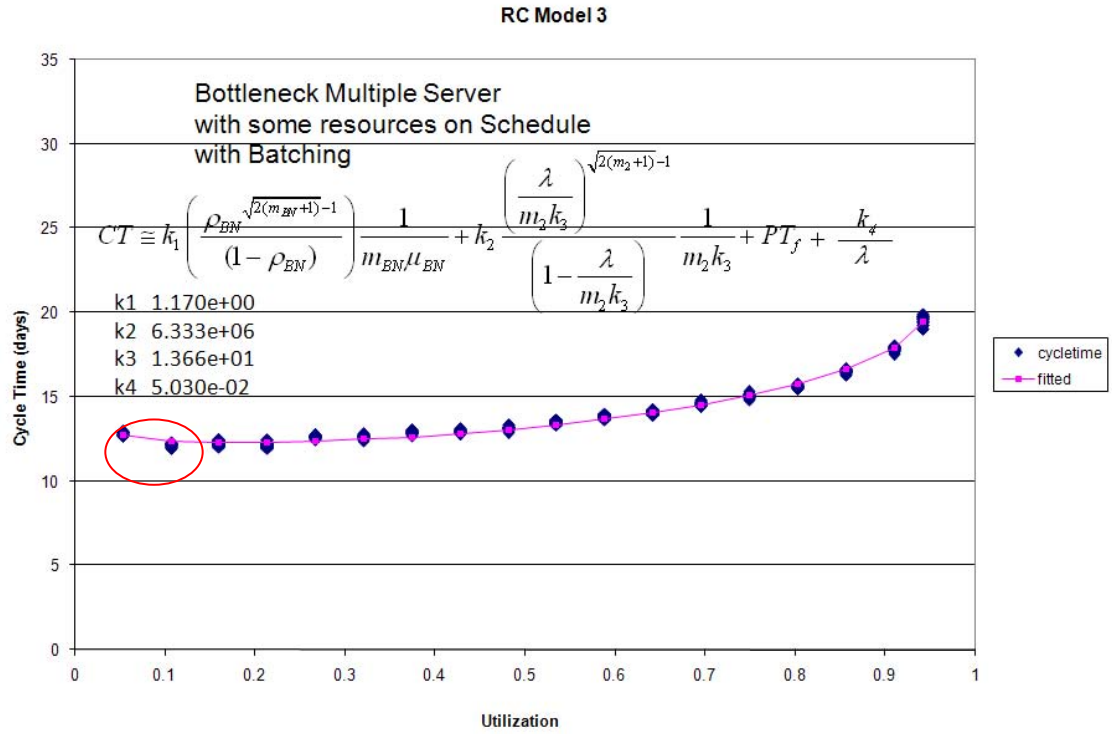


For comparison, the following is the same fit without the k4/λ term:

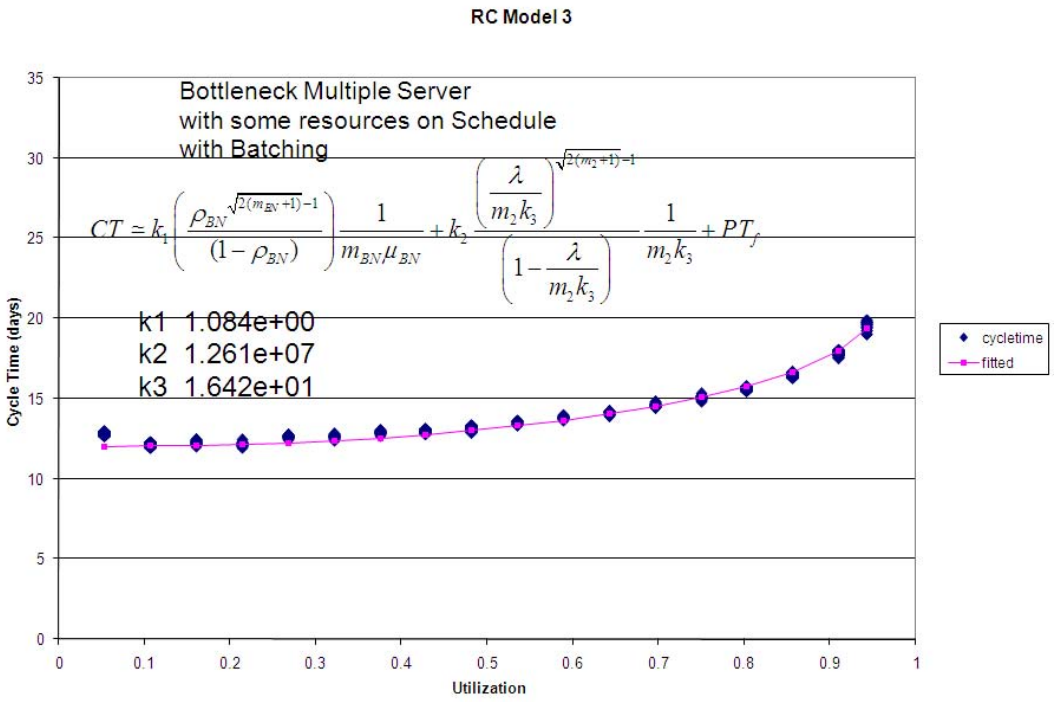
1.2) Model 3



We also tested for G/G/m based approaches for model 3 with the k4/λ term and the results are as follows:



For comparison, this is our earlier G/G/m approach fit without the k4/λ term.



From these two examples, it appears that the batching term with k_4 can make an improvement in the cycle time estimate.

During our earlier investigations we collected data for the low utilizations to estimate PT_f, however, since this data is not of any practical use and has strong batching effects requiring the k4 term, we have truncated the data collected to 20% - 90% range only for our final results.